

В. В. Бондарева (v_bond@rambler.ru),
канд. техн. наук, доцент, доцент кафедры
информационно-вычислительных систем

В. В. Еремин (eremin.v@gmail.com),
магистрант

Белорусский торгово-экономический
университет потребительской кооперации
г. Гомель, Республика Беларусь

WEB-ТЕХНОЛОГИИ ПЕРЕДАЧИ И ОБРАБОТКИ ИНФОРМАЦИИ

Рассматриваются методы передачи и обработки данных от клиента к северу и от сервера к клиенту в сети «Интернет», особенности использования статических и динамических страниц, возможности применения клиентских и серверных сценариев для решения различных задач при формировании информационного наполнения Web-страниц.

Examines methods of transmission and processing of data from the client to the north and from the server to the client over the Internet, features of use of static and dynamic pages, the possibility of using the client and server scenarios for various tasks in the formation content of Web-pages.

Наиболее популярным средством доступа к информации в сети «Интернет» на сегодняшний день является служба World Wide Web (WWW). В ее основе лежит идея распределенной гипермедийной информации. World Wide Web – это название глобальной гипертекстовой системы, которая для передачи информации использует Интернет. Современные гипертекстовые информационные системы условно можно представить в виде совокупности нескольких компонентов:

- системы хранения гипертекстовых объектов;
- системы отображения гипертекстовых объектов;
- системы подготовки гипертекстовых объектов;
- системы программирования просмотра совокупности гипертекстовых объектов.

Технология WWW только к 1996 г. получила законченный, функционально полный вид. Первыми были разработаны системы хранения и просмотра (1989–1991 гг.), которые продолжают развиваться и в настоящее время. После 1990 г. стали появляться первые системы подготовки документов. Наконец, в 1995 г. были предложены первые языки управления сценариями просмотра.

Явная навигация в Web позволяет добраться до самых разнообразных информационных серверов, содержащих специально подготовленную информацию. Основные объемы информации, циркулирующей в сети «Интернет», передаются от серверов на клиентские компьютеры. Как правило, информация обрабатывается перед ее отображением в программах просмотра, причем обработка может выполняться как на сервере, так и на клиентском компьютере.

После того как клиентский компьютер передает с помощью протокола HTTP серверному компьютеру запрос на получение страницы, сервер передает в ответ весь текст, обнаруженный в требуемом файле. Вслед за тем, как данные страницы поступают на клиентский компьютер, браузер выполняет полученный код для вывода страницы на экран, учитывая, какой именно код содержится в файле с данными страницы, каковы предпочтения пользователя, каковы размеры экрана и другие факторы.

В случае использования статических страниц, написанных на языке HTML, содержимое файла на сервере полностью совпадает с исходным кодом страницы, поступившим на клиентский компьютер, т. е. не изменяется при просмотре. Такой подход предоставляет следующие преимущества:

- этот код может быть правильно отображен в любом браузере;
- может быть легко обеспечено адекватное отображение этого кода в устройствах многих других типов;
- каждый запрос выполняется быстро, с использованием минимальных ресурсов;
- язык HTML является простым в изучении, а автоматическое формирование этого кода можно обеспечить очень легко;
- разработчики программ для Web имеют возможность быстро вносить небольшие изменения в отдельные страницы.

Безусловно, статический код HTML имеет также свои недостатки:

- при использовании этого кода трудно управлять процессом создания, оформления и компоновки страницы;

- по мере увеличения количества страниц сложности, связанные с применением статического кода HTML, стремительно возрастают;
- такой код не обеспечивает применения достаточно выразительных интерактивных средств;
- при использовании статического кода HTML затрудняется задача включения значимых метаданных о странице;
- применение статического кода HTML не позволяет успешно отображать быстро изменяющееся информационное наполнение или обеспечивать персонализацию (учет потребностей конкретного пользователя).

Для преодоления указанных ограничений были разработаны многочисленные дополнительные технологии, включая языки JavaScript и VBScript, каскадные таблицы стилей, применяемые в клиентском коде апплеты Java, а также средства поддержки серверных сценариев, предоставляющие такие возможности, как подключение к базе данных. Эти технологии обеспечивают дополнительную функциональность Web-страниц и позволяют создать динамические страницы.

Динамические HTML-страницы позволяют решать более широкий круг задач по сравнению со статическими. В качестве динамических должны рассматриваться почти любые страницы, код которых содержит какие-либо конструкции, отличные от конструкций простого, стандартного кода HTML. В клиентском коде под динамическим информационным наполнением могут подразумеваться мультимедийные презентации, прокручиваемые заголовки, автоматически обновляющие сами себя страницы или элементы, способные исчезать и появляться, т. е. обеспечение пользовательского интерфейса. В серверном коде динамическое информационное наполнение подразумевает содержимое страницы, наполнение которого происходит на основании запроса на передачу страницы, т. е. применяется для обработки данных.

В последнее время средства поддержки серверных сценариев находят все более широкое распространение, поэтому в распоряжение пользователей Web все чаще поступают динамически сформированные HTML-страницы.

Наиболее часто дополнения к простому коду HTML реализуются после поступления страницы в клиентское приложение. К этим дополнениям относятся расширения, обеспечивающие лучшее форматирование, например, каскадные таблицы стилей, средства Dynamic HTML, а также клиентские языки сценариев, такие как JavaScript и VBScript, апплеты Java и объекты Flash. Средства поддержки всех этих технологий, как правило, встроены в программное обеспечение современного браузера.

С помощью клиентских сценариев можно решить множество задач:

- обработка событий, инициированных пользователем (щелчок кнопкой мыши, нажатие клавиши и др.);
- обработка данных, введенных пользователем в форму, перед их отправкой на сервер;
- изменение параметров отдельных элементов документа и браузера (изменение значений атрибутов, стилевых параметров и даже тегов);
- запись и чтение cookie;
- управление процессами во времени и т. д.

Достоинством клиентских технологий является высокая скорость реакции на действия пользователей, так как не требуется передавать данные на сервер.

Клиентские технологии имеют ряд недостатков, например их выполнение полностью зависит от браузера, а между браузерами разных производителей могут быть большие отличия. Клиентские технологии не позволяют выполнять какие-либо задачи, требующие подключения к серверу базы данных.

Серверные сценарии выполняют задачи, с которыми достаточно сложно справиться с помощью клиентских сценариев:

- прием и обработка данных, введенных посетителями Web-страницы;
- сохранение данных в файлах и базах данных, расположенных на стороне сервера;
- передача данных по электронной почте;
- модификация графических изображений;
- поддержка работы счетчика посещения страниц, гостевой книги, форума, интернет-магазина, защиты страниц сайта паролем и т. д.

Для поддержки серверных сценариев необходимо установить на компьютер специальное программное обеспечение (Web-сервер, интерпретатор языка, систему управления базами данных и др.). Такая машина обеспечивает синтаксический анализ и интерпретацию страниц, написанных на определенном языке.

Для создания серверных сценариев могут быть использованы языки SSI, PHP, Perl, Python, ASP и т. д.

В состав Windows входит сервер Internet Information Services (IIS), который обеспечивает поддержку Web, FTP, FrontPage, транзакций, Active Server Pages (ASP) и подключений к базам данных. Однако чаще используется сервер Apache, который позволяет создавать динамические сайты с использованием различных языков программирования (например, PHP).

PHP – это серверный язык сценариев, который может быть внедрен в код HTML для развертывания в браузере. Преобразование программы на языке PHP в HTML производится на сервере, по сети передается готовый для просмотра код. Благодаря выполнению на сервере, программа на языке PHP может отображать данные, которые динамически изменяются (например, текущее состояние прайс-листа или текущие котировки курсов акций).

В Web имеется одна возможность, которую можно использовать для передачи информации, – это формы. Язык HTML предоставляет возможность создавать на страницах формы для ввода пользовательской информации. При навигации по страницам часто можно встретить некоторую форму, которую необходимо заполнить. Форма, заполненная в клиентской части системы, поступает на обработку соответствующему программному сценарию, связанному с данной формой в серверном узле Web.

Существует два метода передачи информации от формы к обработчику (GET и POST). Введенные пользователем данные должны быть обработаны в обработчике формы, в качестве которого может выступать PHP-код. Конкретный метод указывается в атрибуте METHOD формы, а имя обработчика в атрибуте ACTION (например: `<form action = "http://localhost/base.php" method = "get">`).

Механизмом передачи информации между формой и обработчиком событий являются суперглобальные массивы `$_GET`, `$_POST`, `$_COOKIE`, `$_ENV`, `$_SERVER` и др. Данные из полей ввода форм автоматически присваиваются элементам этих массивов. В качестве индексов элементов массивов используются имена элементов ввода, задаваемые в их свойстве NAME.

Метод GET предусматривает передачу параметров в составе строки запроса, которая представлена в формате универсального идентификатора ресурса URL (например: <http://localhost/base.php?T1=bteu&Submit=Select>).

Эта строка будет использована браузером как адрес страницы, на которую следует перейти. Знак вопроса «?» после имени файла обозначает начало передачи переменных методом GET. После него следуют пары «имя=значение», разделенные символом амперсанда «&». Так браузер автоматически присоединяет данную информацию к URL при отправке запроса Web-серверу.

Получить доступ к GET-параметрам можно через суперглобальный массив `$_GET`, указав в квадратных скобках после него имя параметра. Передача данных методом GET не всегда является удобной по следующим причинам:

- пользователь может видеть значение параметров и легко подделывать их в строке запроса;
- объем передаваемой информации через GET-параметры ограничен (как правило, 8 Кбайт).

Существует еще один способ передачи данных – через тело документа. Для этого предназначен метод POST. Для получения POST-параметров необходимо использовать суперглобальный массив `$_POST`.

У метода POST также есть свои особенности:

- Объем передаваемой информации через POST-параметры также ограничен (как правило, 8 Мбайт).

- Страницы, загруженные с его помощью, невозможно записать в закладки браузера, а страницы, загруженные с помощью метода GET, можно, так как они содержат всю необходимую информацию в URL. Кроме того, страницы, работающие с помощью метода POST, не могут быть проиндексированы поисковыми машинами.

- Несмотря на то, что информация не видна пользователю, она не шифруется и может быть доступна хакерам. Для того чтобы гарантировать безопасность информации, необходимо использовать защищенное соединение с безопасным сервером.

Решение о том, какой метод можно использовать в форме, зависит от задач, решаемых с помощью данной формы.